Certification Concepts for AI/ML Systems

Dr. Guillaume Brat
Robust Software Engineering
Intelligent Systems Division
NASA Ames Research Center

RSE NASA

Research context

- Research done at NASA Ames Research center under Technical Challenge 4 (TC 4) in the System-wide Safety (SWS) project in the Airspace Operations & Safety program (AOSP) in the Aeronautics Research & Mission Directorate (ARMD).
    - Aligned with ARMD Thrust 6: Assured autonomy
    - Deon by Robust Software Engineer group at NASA Ames Research center

SWS Research Portfolio

Operational Safety (Thrust 5)

TC-1: *Predictive Terminal Area Risk Assessment*

TC-2: *IASMS SFC Development for Emerging Operations*

Current Day

Near Future

TC-5: *Safety Demonstrator Series for Operational IASMS*

Transformed NAS

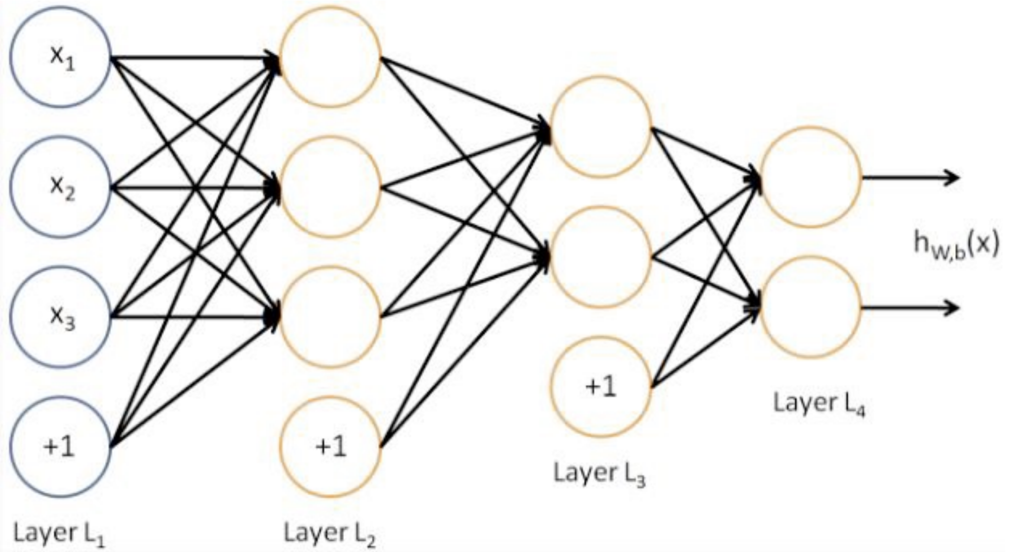
TC-3: *V&V for Commercial Operations*

TC-4: *Complex Autonomous Systems Assurance*

Design Safety (Thrust 6)

SWS TC4 major milestones

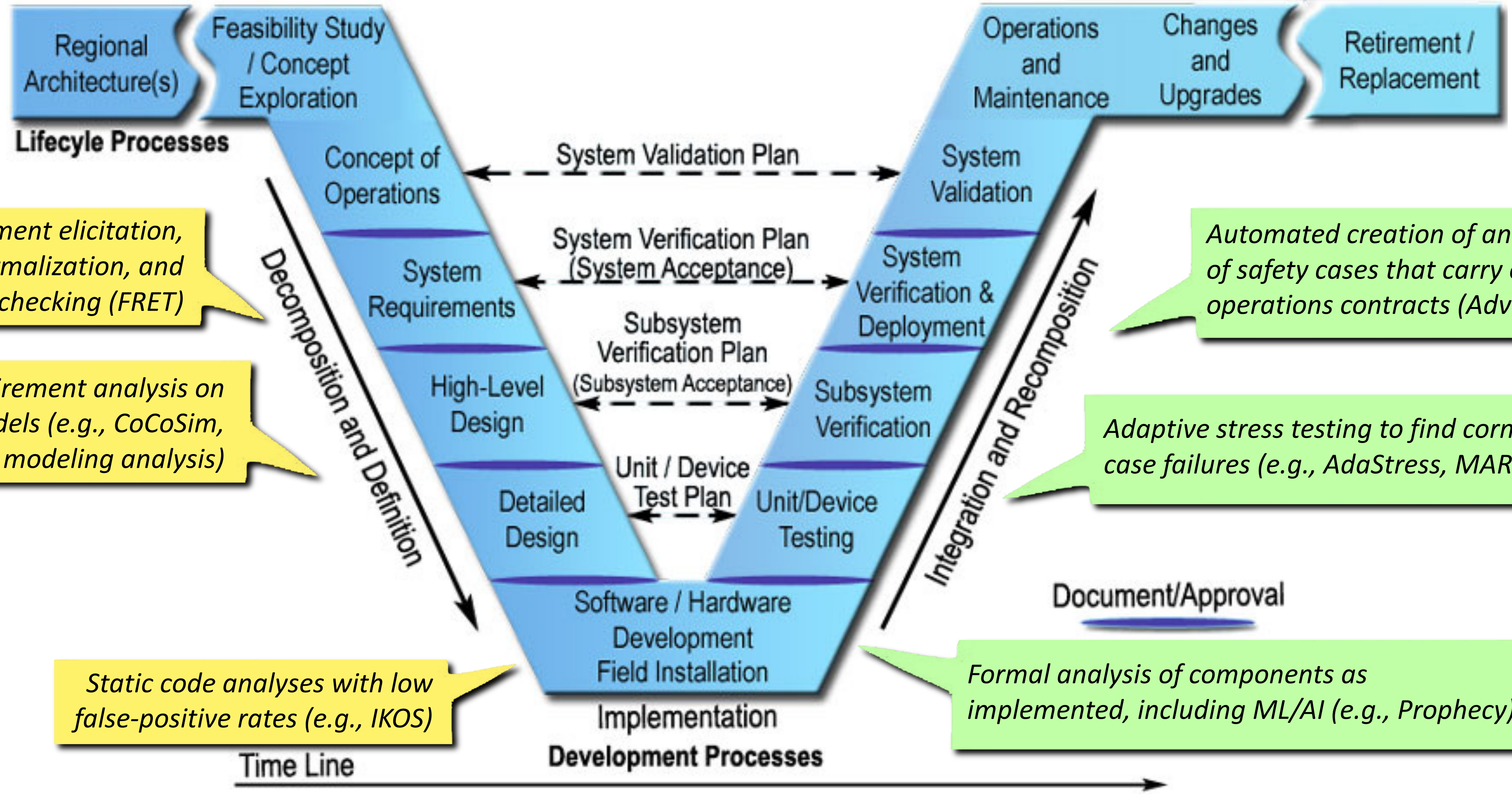
- 2022: Delivery of draft evidence and recommendations for the robustness of failover plans and the use of run-time monitoring
- 2023: Demonstration of algorithms for chec
systems relying on untrusted components fc
operations and autonomous drone flight op
- 2024: Preliminary certification process for M
aerospace systems



Major research themes

- Improving safety and risk assessment as early as possible in the lifecycle
- Elicitation and formalization of requirements to facilitate traceability throughout the lifecycle, especially when formal methods are used
- Algorithms, tools and techniques for the V&V of ML-enabled systems
- Advanced testing
- Use of runtime monitoring to ease use of untrusted components
- Contribution to draft regulatory standards and assistance in producing and presenting certification evidences

Some RSE tools on the V



Operational run-time analysis (R2U2, CoPilot, MESA)

Requirement elicitation, formalization, and consistency checking (FRET)

Automated creation of and visualization of safety cases that carry over into operations contracts (AdvoCATE)

Safety requirement analysis on system models (e.g., CoCoSim, SAVI modeling analysis)

Adaptive stress testing to find corner case failures (e.g., AdaStress, MARGInS)

Static code analyses with low false-positive rates (e.g., IKOS)

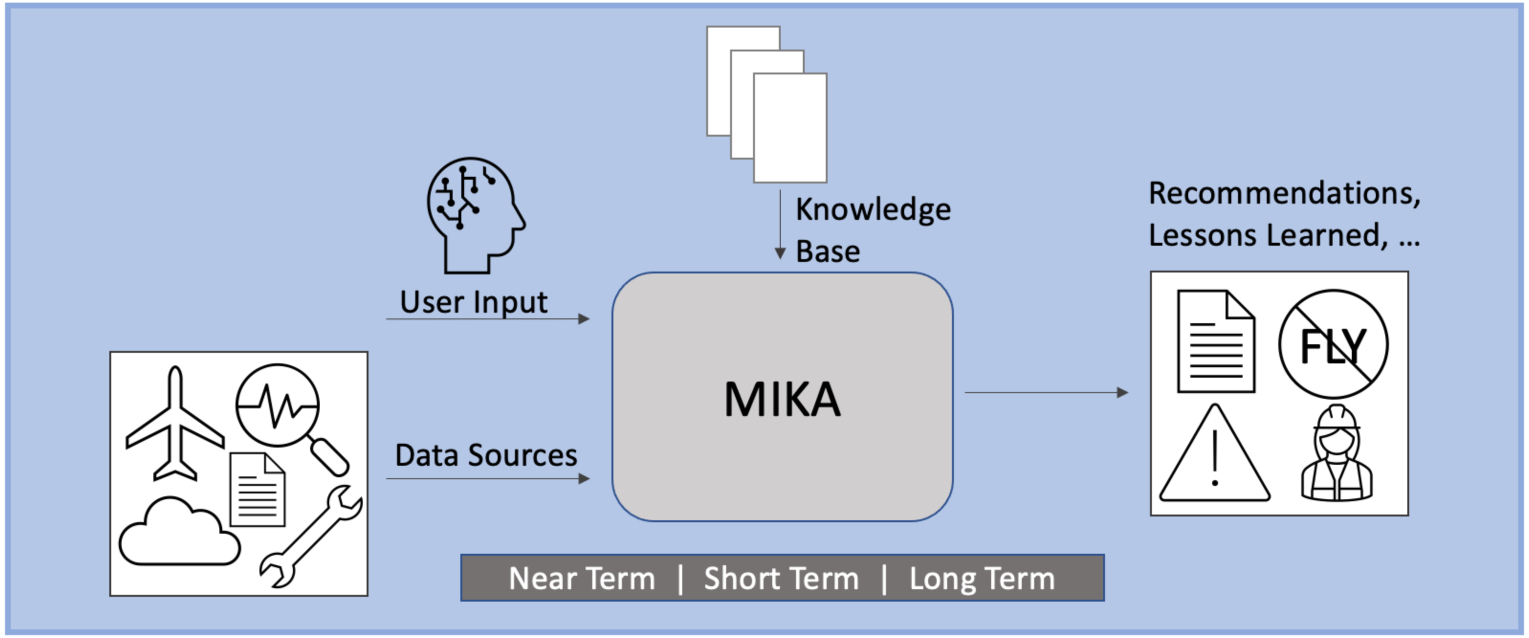
Formal analysis of components as implemented, including ML/AI (e.g., Prophecy)

- Improving safety and risk assessment as early as possible in the lifecycle
- Elicitation and formalization of requirements to facilitate traceability throughout the lifecycle, especially when formal methods are used
- Algorithms, tools and techniques for the V&V of ML-enabled systems
- Advanced testing
- Use of runtime monitoring to ease use of untrusted components
- Contribution to draft regulatory standards and assistance in producing and presenting certification evidences

NLP-based risk assistance

- Hazard analysis frequently relies on historical knowledge of failure modes for prevention.
- Advances in natural language processing (NLP) enable extraction of knowledge stored in large, unstructured sets of documents of lessons learned and accident reports.

MIKA: *Manager for Intelligent Knowledge Access*. An assistive knowledge manager for decision support and formulating recommendations in the In-Time Aviation Safety Management System (IASMS).



Resilience analysis

Fmdtool: A tool to analyze system resilience early in design

What is fmdtool?

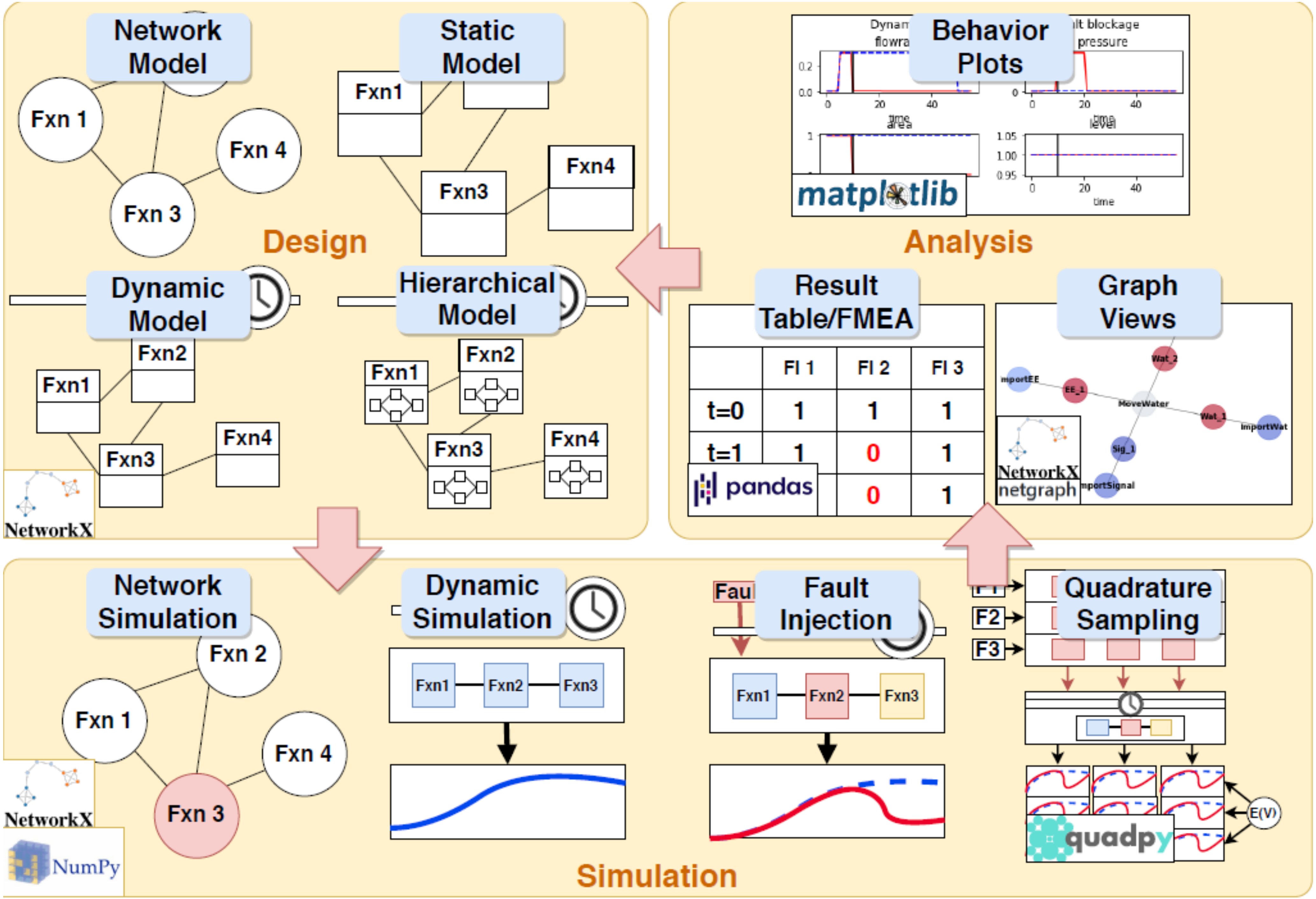
- A modelling framework
- Simulation methods
- Visualizations/Metrics

  …

- A design environment

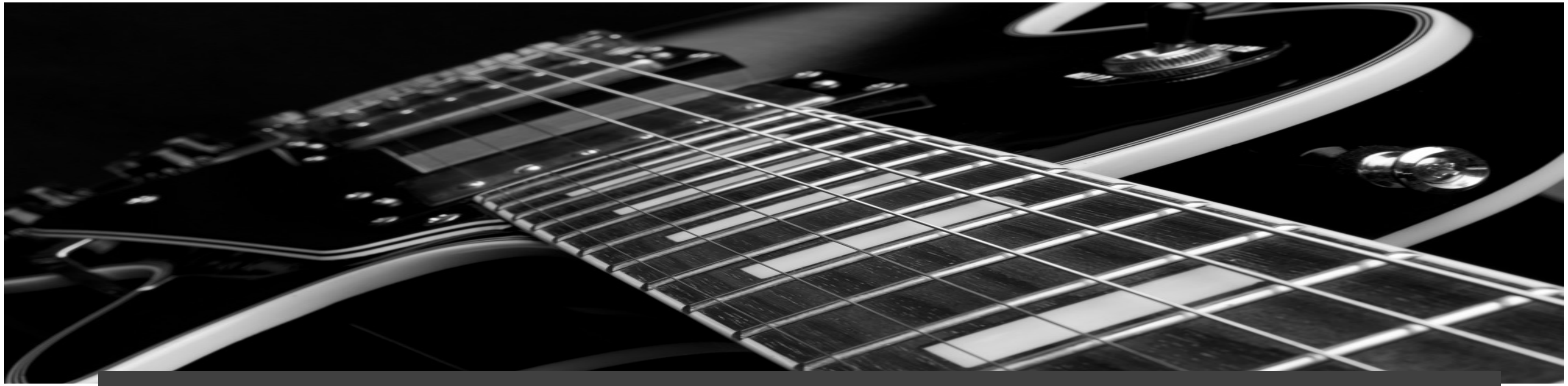
Why use fmdtool?

- Expressive:
  - Undirected graph propagation
    - faults from any component can propagate to any other connected component
  - General model representation—not a strict formalism
    - Behavioral (equations), fault logic (if-else statements), etc.
  - Dynamic simulation needed to **quantify resilience**
- Research-oriented:
  - Written in/relies on the Python stack
  - Open source/free software
- Enables design:
  - Models can be parameterized an optimized!
  - Provides tools to visualize and quantify simulation results



Design
Network Model
Fxn 1
Fxn 3
Fxn 4
Static Model
Fxn1
Fxn3
Fxn4
Dynamic Model
Fxn1
Fxn2
Fxn3
Fxn4
Hierarchical Model
Fxn1
Fxn2
Fxn3
Fxn4
NetworkX
Analysis
Behavior Plots
pressure
time
level
matplotlib
Result Table/FMEA
FI 1
FI 2
FI 3
t=0
1
1
1
t=1
1
0
1
0
1
pandas
Graph Views
Wat_2
ImportEE
EE_1
MoveWater
Wat_1
ImportWat
Sig_1
ImportSignal
NetworkX
netgraph
Simulation
Network Simulation
Fxn 1
Fxn 2
Fxn 3
Fxn 4
NetworkX
NumPy
Dynamic Simulation
Fxn1
Fxn2
Fxn3
Fault Injection
Fxn1
Fxn2
Fxn3
Quadrature Sampling
F2
F3
E(V)
quadpy

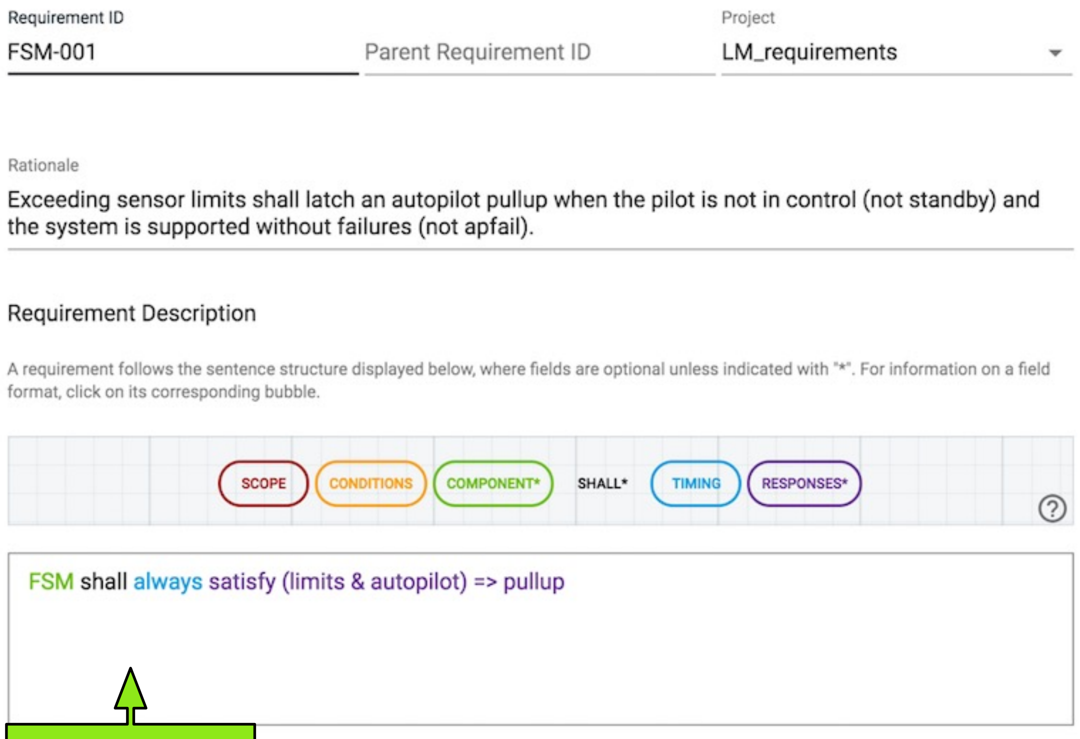
- Improving safety and risk assessment as early as possible in the lifecycle
- Elicitation and formalization of requirements to facilitate traceability throughout the lifecycle, especially when formal methods are used
- Algorithms, tools and techniques for the V&V of ML-enabled systems
- Advanced testing
- Use of runtime monitoring to ease use of untrusted components
- Contribution to draft regulatory standards and assistance in producing and presenting certification evidences



FRET: Formal Requirements Elicitation Tool

- Extensible grammar defines a restricted natural language; requirements made up of fields for scope, conditions, component, timing, response
- Compositional generation of semantics from requirement fields; semantics output to model checkers for consistency analysis
- Explanations of the formal semantics in various forms: natural language, diagrams, interactive simulation
- Connects requirements to Simulink models for verification with Cocosim and Simulink Design Verifier

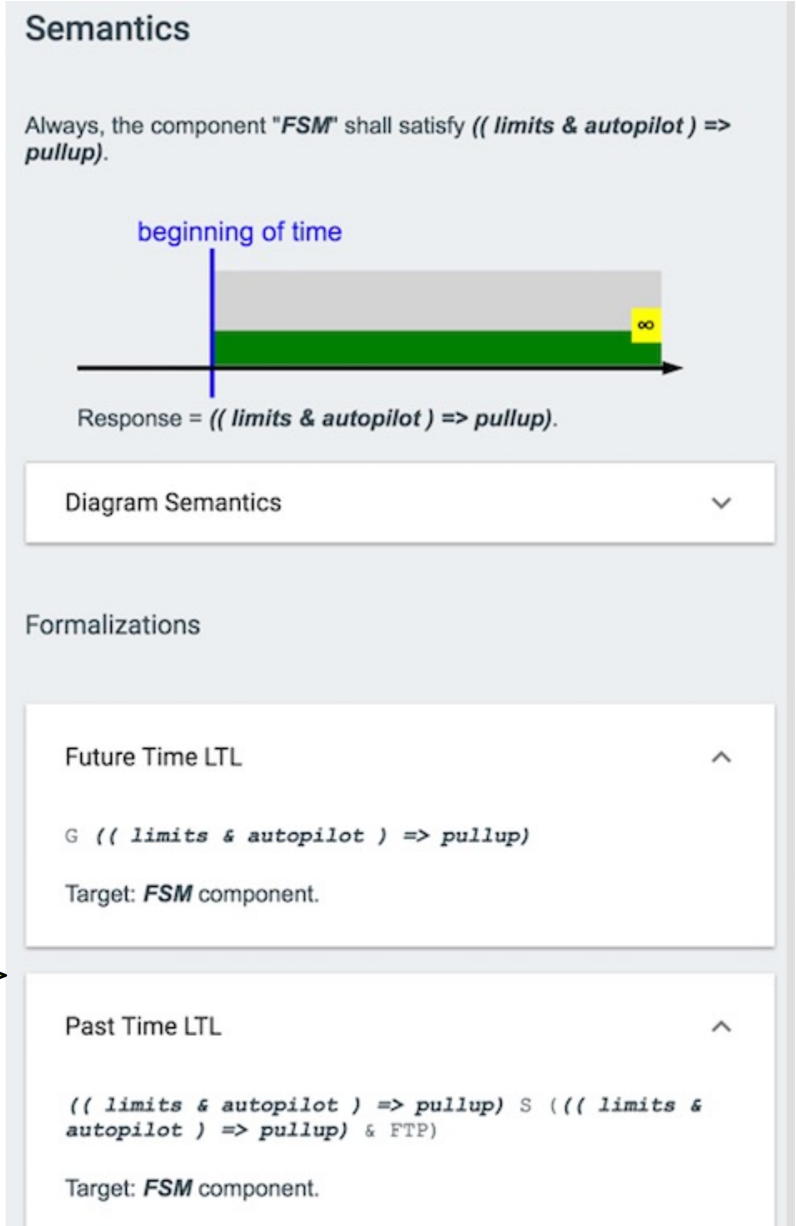
simple requirement in FRET



user types requirement

parser recognizes fields and color codes them dynamically

FRET generates formal semantics



more complex requirement

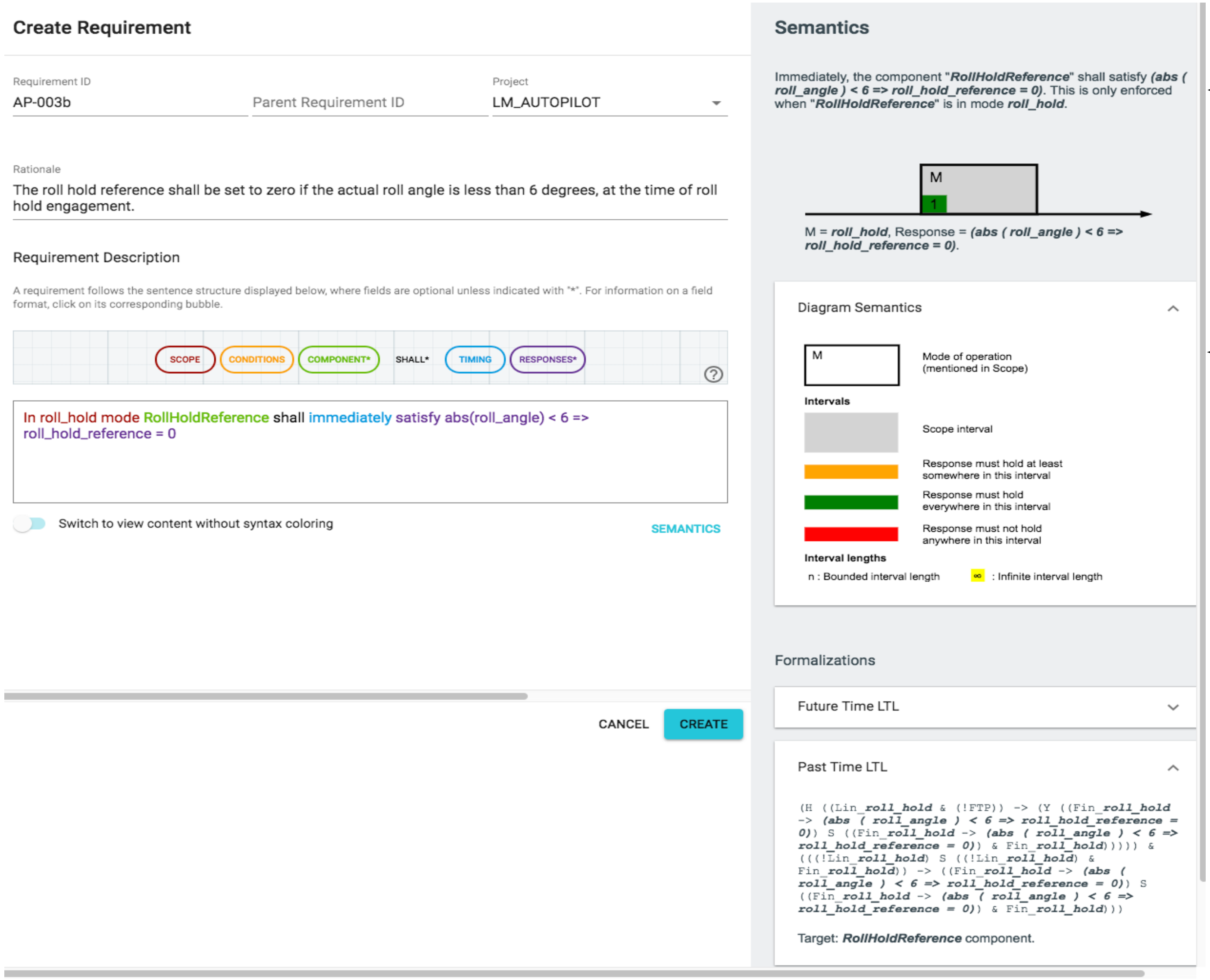


diagram provides intuitive explanation

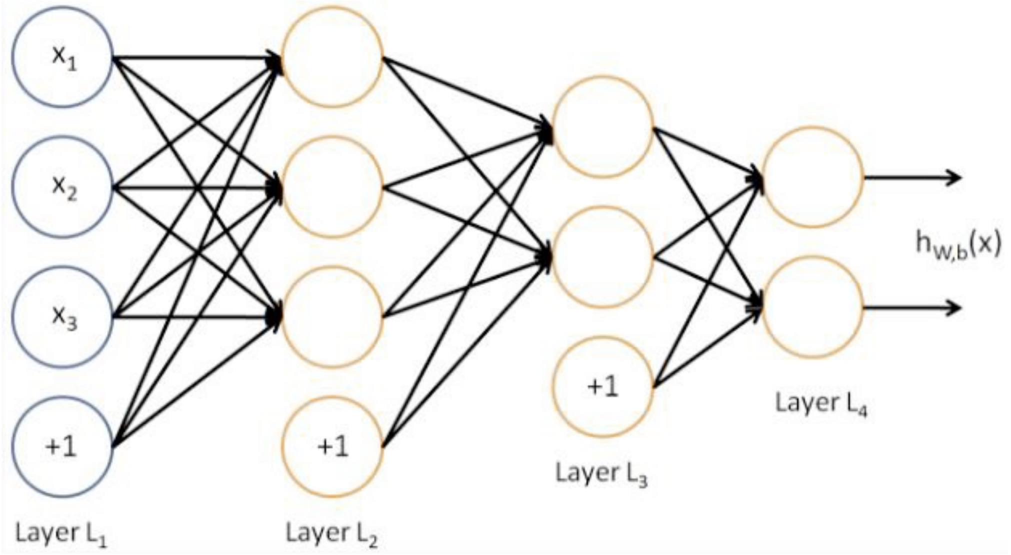
on-demand diagram semantics

formulas are more involved; would be hard to write manually

- Improving safety and risk assessment as early as possible in the lifecycle
- Elicitation and formalization of requirements to facilitate traceability throughout the lifecycle, especially when formal methods are used
- Algorithms, tools and techniques for the V&V of ML-enabled systems
- Advanced testing
- Use of runtime monitoring to ease use of untrusted components
- Contribution to draft regulatory standards and assistance in producing and presenting certification evidences

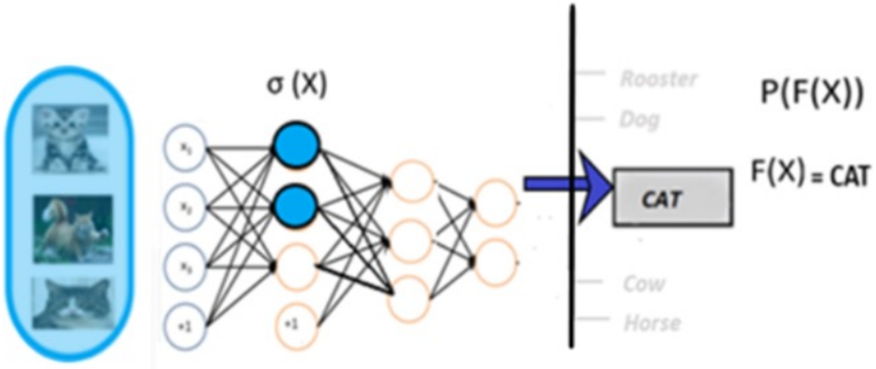
SafeDNN

- Deep Neural Networks (DNNs) have widespread usage, even in safety-critical applications such as autonomous driving
- Develop techniques that aim to ensure that systems that use DNNs are safe, robust and interpretable
- Prophecy: Formal analysis of DNN models to infer properties which could be used for understanding, verifying, debugging and testing



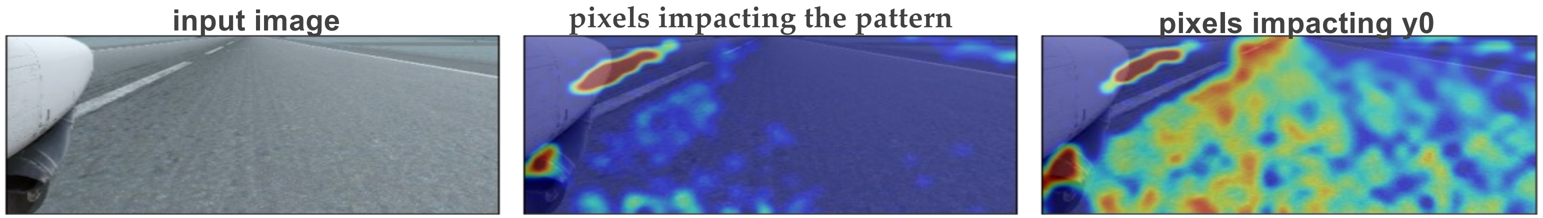
Approach: Prophecy to formally V&V a DNN

- Decompose the complex DNN model into a *set of simple rules*, amenable to analysis
  - Assume-guarantee type rules are inferred from a trained DNN; $\forall x\ \sigma\ (x) \Rightarrow P(F(x))$
  - P is a property of the network function; functional property
  - σ (X) are formal constraints on neurons at inner layers of the network *(neuron activation patterns)*
  - *Prophecy:* Property Inference for Deep Neural Networks (ASE 2019)



Prophecy on taxinet

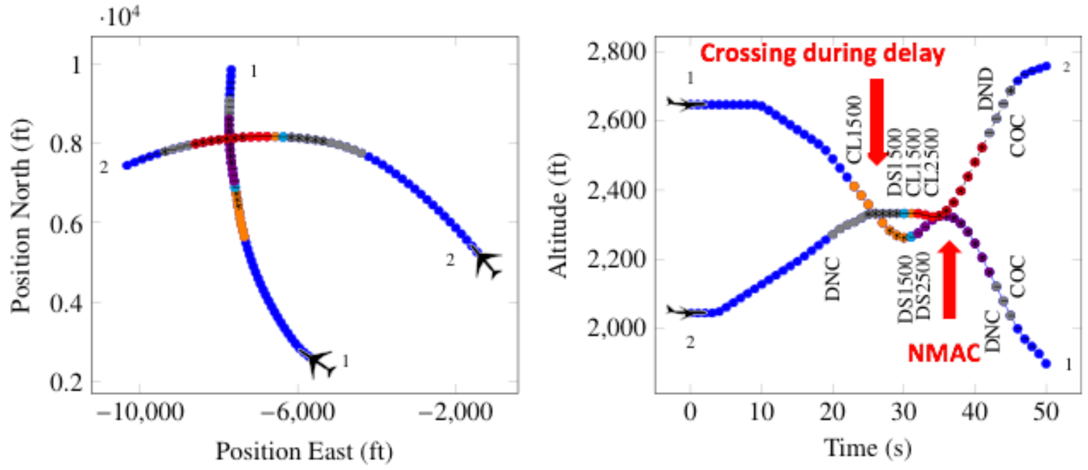
- *Boeing TaxiNet:* CONV network with 24 layers, input is a 360x200x3 image, 5 CONV layers, 5 activation layers and 3 dense layers (100,50,10 eLU neurons respy) before the output layer with 2 outputs
- Prophecy used to extract patterns using a labeled dataset with 13885 inputs
  - Wrt three *correctness properties*; $|y_0 - y_{0ideal}| \leq 1.0$, ,$|y_1 - y_{1ideal}| \leq 5.0$, $|y_0 - y_{0ideal}| \leq 1.0 \wedge |y_1 - y_{1ideal}| \leq 5.0$
  - At each of the three dense layers and all of them together
  - *Patterns for satisfaction* (396 patterns for class 1), *patterns for violation* of the correctness properties (418 patterns for class 0)
- *Tiny Taxinet [3]:* Smaller network takes in a down-sampled version of the image (128 pixels), 3 dense layers (16,8,8 ReLU neurons respy) and output layer with 2 outputs
- Prophecy used to extract patterns using a labeled dataset with 51462 inputs
  - Wrt three *safety properties*; $|y_0| \leq 10.0$, $|y_0| \leq 8.0$, $|y_0| \leq 5.0$
  - At each of the three dense layers and all of them together, patterns for satisfaction and violation of the safety properties were extracted



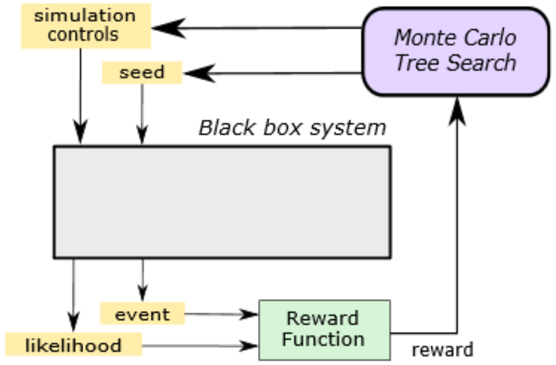
- Improving safety and risk assessment as early as possible in the lifecycle
- Elicitation and formalization of requirements to facilitate traceability throughout the lifecycle, especially when formal methods are used
- Algorithms, tools and techniques for the V&V of ML-enabled systems
- Advanced testing
- Use of runtime monitoring to ease use of untrusted components
- Contribution to draft regulatory standards and assistance in producing and presenting certification evidences

Adaptive Stress testing

- <u>Motivation and Objective</u>
    - AdaStress is a software package for an accelerated simulation-based stress testing method for finding the most likely path to a failure event
    - Motivated by the needs of the ACAS-X project at the FAA, where ACAS-X is the next generation of on-board collision avoidance systems.

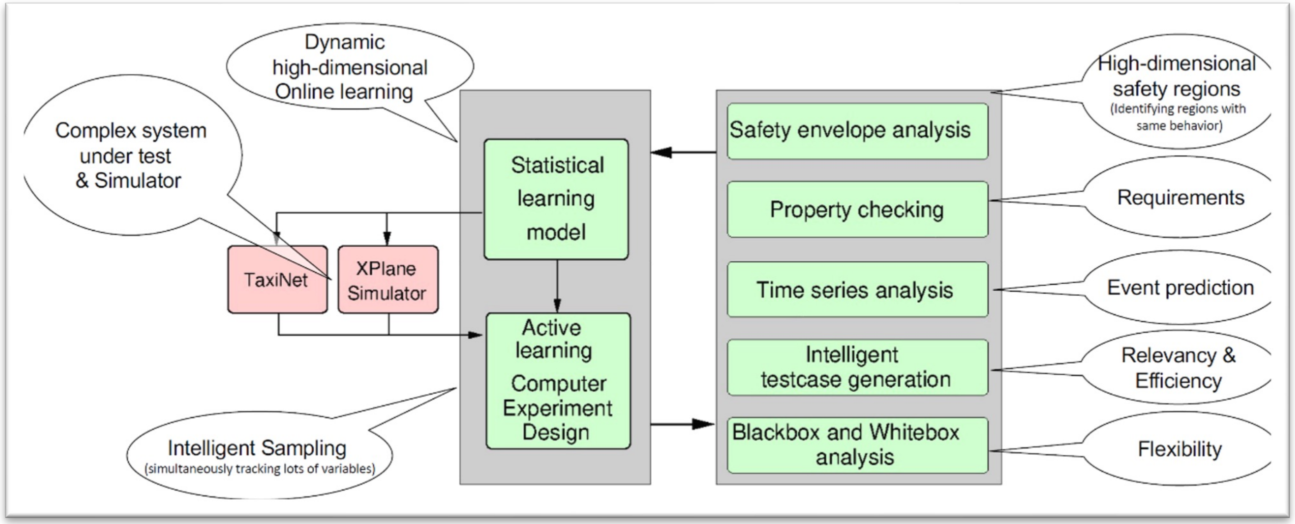


- <u>Approach</u>
    - Turn AI on AI
        - Use Reinforcement Learning techniques to drive testing towards rare failure events
    - Provide explanation capabilities to explain why the failure is a failure using grammar-based decision trees.

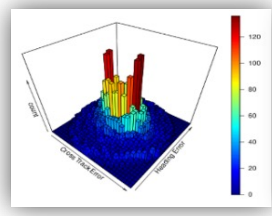


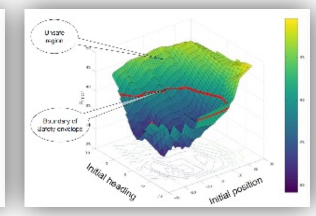
SysAI

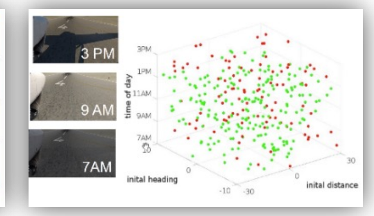
- <u>Motivation and Objective</u>
  - Provide advanced capabilities that support understanding the system behavior in nominal and off-nominal situations.
  - MARGInS is a framework that enables the user to create customized machine learning and statistical tool chains for analyzing and predicting the behavior of a complex, hybrid system.



- <u>Approach</u>
  - MARGInS contains set of machine learning and statistical algorithms for multivariate clustering, treatment learning, critical factor determination, time-series analysis, event prediction, and safety-boundary detection and characterization.
  - Key benefits:
    - **Supports system testing**
    - **Configurable** – find novel features in test suites, determine classes of behavior, propose new experiments that can efficiently explore the boundaries between classes of behavior, and to create visualizations and reports.

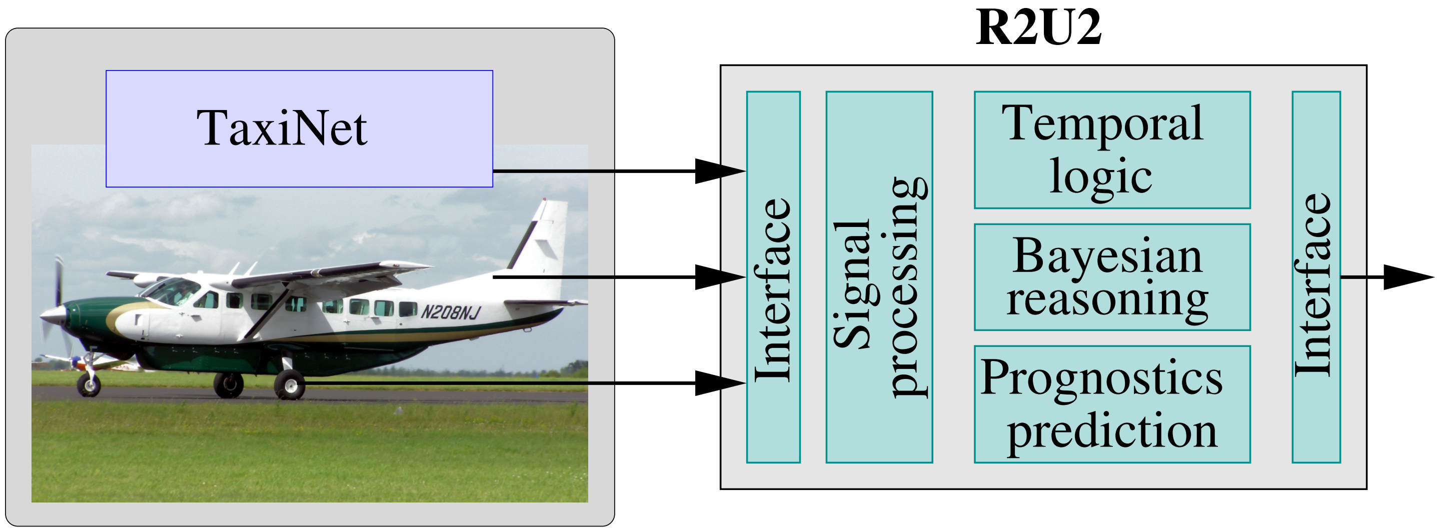






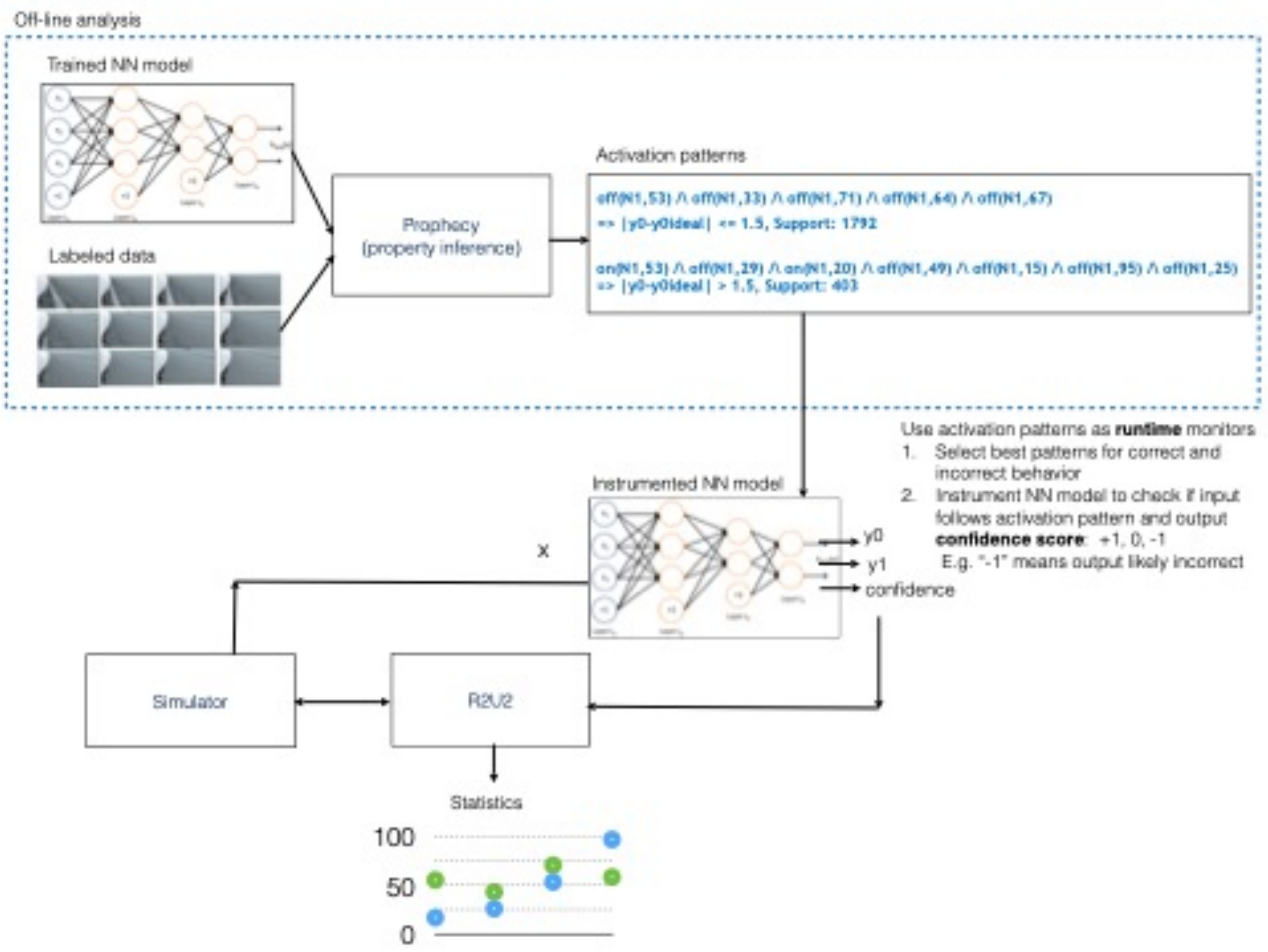
- Improving safety and risk assessment as early as possible in the lifecycle
- Elicitation and formalization of requirements to facilitate traceability throughout the lifecycle, especially when formal methods are used
- Algorithms, tools and techniques for the V&V of ML-enabled systems
- Advanced testing
- Use of runtime monitoring to ease use of untrusted components
- Contribution to draft regulatory standards and assistance in producing and presenting certification evidences

R2U2: runtime monitoring



R2U2 is a run-time monitoring and V&V tool that combines *Metric Temporal Logic* observers, *Bayesian Network* reasoners, and *model-based prognostics*.

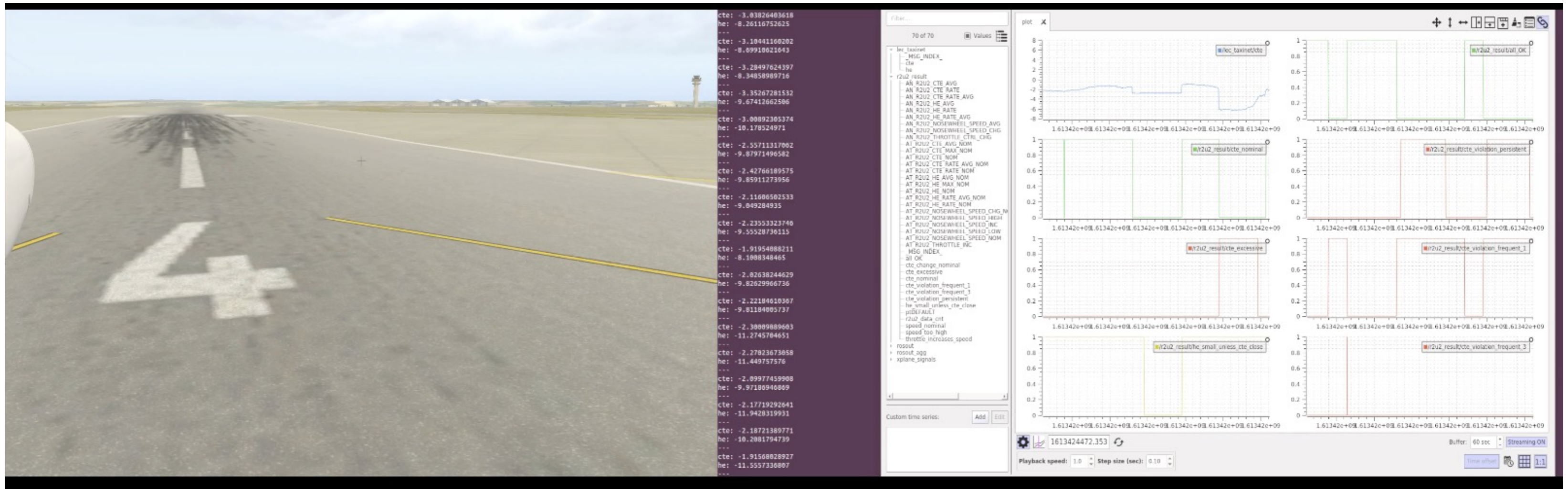
Prophecy-R2U2 integration



R2U2 capabilities

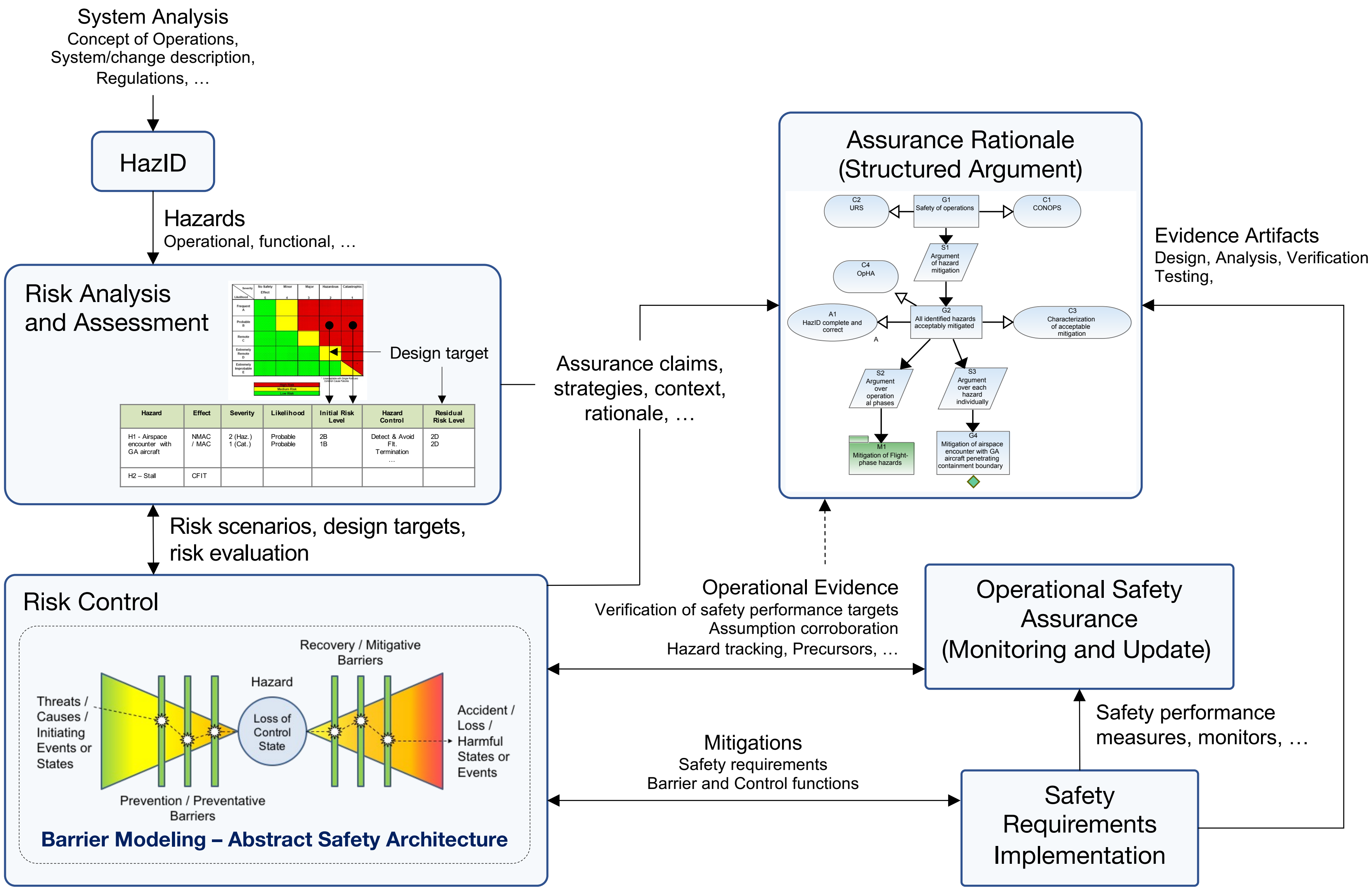
- Signal Processing
- Past Time Temporal Logic
- Future Time Temporal Logic
- Bayesian Reasoning
- Prognostics

- safety monitoring
- performance monitoring
- security monitoring
- failure diagnosis
- prognostics
- autonomous decision making

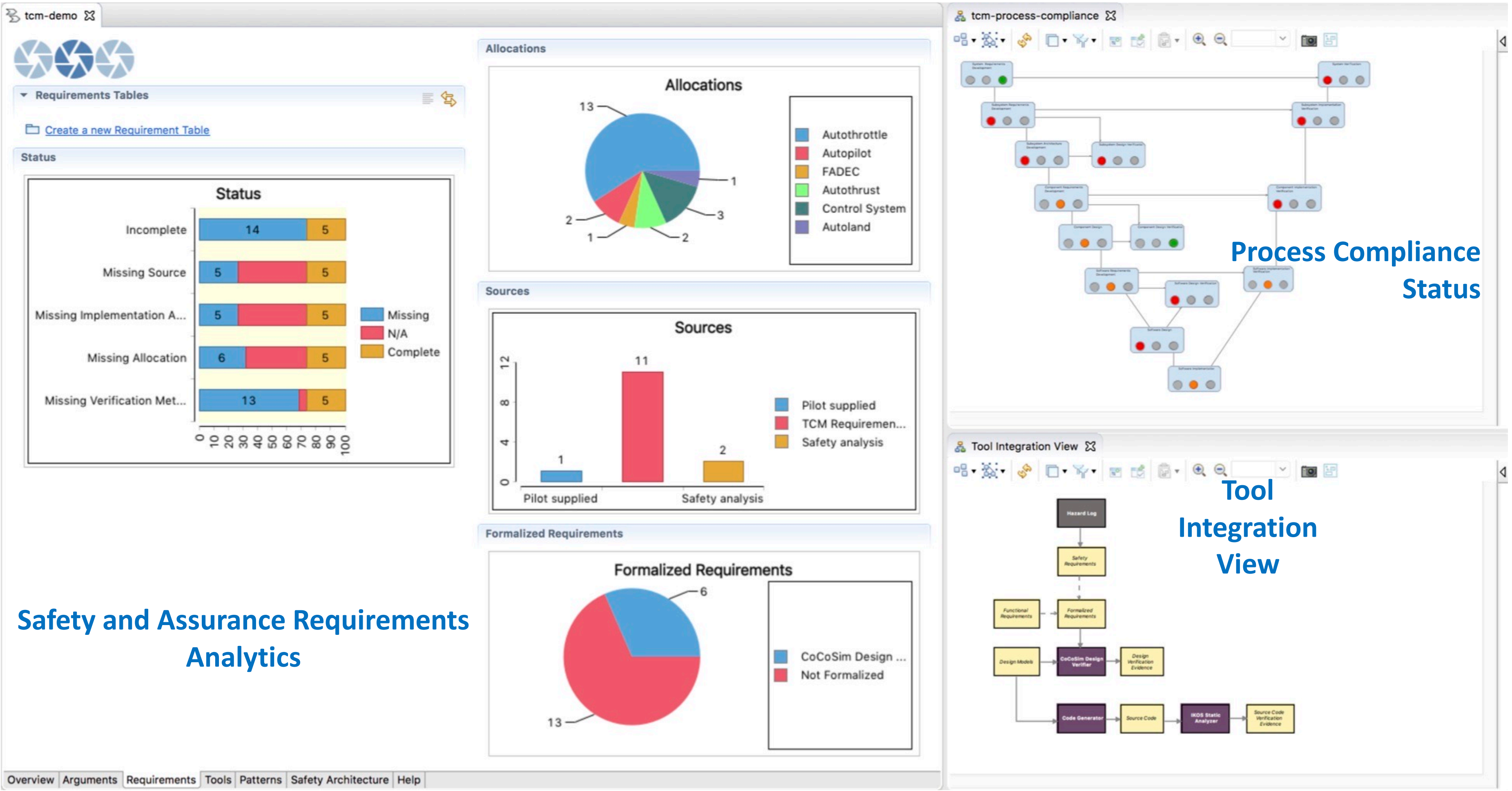


- Improving safety and risk assessment as early as possible in the lifecycle
- Elicitation and formalization of requirements to facilitate traceability throughout the lifecycle, especially when formal methods are used
- Algorithms, tools and techniques for the V&V of ML-enabled systems
- Advanced testing
- Use of runtime monitoring to ease use of untrusted components
- Contribution to draft regulatory standards and assistance in producing and presenting certification evidences

AdvoCATE: Safety Risk Management



Dynamic Safety Case Dashboards



Conclusions

- The RSE group at NASA Ames Research Center is working on tools and techniques for the assurance and certification of increasingly autonomous (e.g., ML-enabled) systems in aviation.
- Major research themes are:
  - Improving safety and risk assessment as early as possible in the lifecycle
  - Elicitation and formalization of requirements to facilitate traceability throughout the lifecycle, especially when formal methods are used
  - Algorithms, tools and techniques for the V&V of ML-enabled systems
  - Advanced testing
  - Use of runtime monitoring to ease use of untrusted components
  - Contribution to draft regulatory standards and assistance in producing and presenting certification evidences
- POC: Guillaume Brat, guillaume.p.brat@nasa.gov

Tools	Description	Availability	Technical POC	POS Email
AdvoCATE	Assurance case automation toolset	Open Source	Ewen Denney	ewen.w.denney@nasa.gov
AdaStress	Adaptive stress testing	Open Source	Adrian Agogino	adrian.k.agogino@nasa.gov
CoCoSim	Simulink model analyzer	Open Source	Andreas Katis	andreas.katis@nasa.gov
Drishti	Compliance Assistant	Not available Yet	Nija Shi	nija.shi@nasa.gov
Fmdtool	System resilience analysis	Open Source	Daniel Hulse	daniel.e.hulse@nasa.gov
FRET	Requirement elicitation and analysis	Open Source	Anastasia Mavridou	anastasia.mavridou@nasa.gov
IKOS	Static code analysis for C/C++	Open Source	Guillaume Brat	guillaume.p.brat@nasa.gov
MARGInS	ML/statistical libraries for system testing	Usage Agreement	Carlos Paradis	carlos.v.paradis@nasa.gov
MIKA	NLP-based risk analysis	Not available Yet	Hannah Walsh	hannah.s.walsh@nasa.gov
Prophecy	Formal analysis of Neural Networks	Not available yet	Corina Pasareanu	corina.s.pasareanu@nasa.gov
RACE	Runtime for Airspace Concept Evaluation	Open Source	Peter Mehlitz	peter.c.mehlitz@nasa.gov
R2U2	Vehicle-level run-time analysis	Usage Agreement	Johann Schumann	johann.m.schumann@nasa.gov
SysAI	ML/statistical libraries for system testing	Not available yet	Yuning He	yuning.he@nasa.gov